GCSE ICT
Practical Course Pack 2

Steve McWeeney

ISBN-10: 1 84070 185 4
ISBN-13: 978 1 84070 185 2

Published by Pearson Publishing 2002
Revised 2006

Pearson Publishing, Chesterton Mill, French's Road, Cambridge CB4 3NP Tel 01223 350555 Fax 01223 356484
www.pearsonpublishing.co.uk

Students' introduction to ICT practical work

In practical work you will use computer systems to produce solutions to problems. A problem could be something as simple as writing a letter or producing a poster for an event. It could be more complicated, like working out the profit for a business month by month or producing a set of address labels for customers who live in a particular postcode area.

An ICT problem is simply a job that needs to be done, and for which a computer system is an appropriate tool.

To solve a problem properly, you will need to plan your work. If you just rush to a computer and start work, you will find that your solutions are ineffective and do not actually do the job properly. Part of your ICT practical work will involve learning how to plan, so that you end up with a good solution to the problem that you are solving.

ICT professionals spend much of their time solving problems for other people. They need to find out from their client exactly what is required. Only then can they begin to plan out the solution. Once the solution has been produced, they need some way of agreeing with the client that it really does the job. Because of all these considerations, they need to break down the process of solving a problem into different stages. These are explained below.

Stages of solution

There are five stages in solving an ICT problem. They are as follows:

Stage	What happens	Answering the question
Analysis	Find out what the problem is.	What needs doing?
Design	Plan a solution to the problem you have identified.	How will I do it?
Implementation	Produce your solution, following the design plan.	
Testing	Test that your solution works as designed.	Does it work?
Evaluation	Measure how well your solution fits as a solution to the problem.	Does it do the job?

These are often called the five stages of the system life cycle. Four of the stages answer questions. The last two questions are similar; they are to do with whether or not the solution works. The testing stage is checking to see if your design has been implemented correctly (ie whether things actually happen as planned). The last stage – evaluation – is checking that the solution actually solves the problem. It is possible that your implementation works exactly as planned, but that your design does not actually meet the client's needs. We will now look at each of the five stages in more detail.

Analysis

The analysis stage of solving a problem involves talking to the client and finding out what the problem is. The client is the person who has the problem you are solving. At the end of this process you should be able to write a clear description of what the client needs. This is called the **problem specification**.

The problem specification might be just a single sentence for a simple problem. In many cases, it will seem so obvious that you might think it is silly to write it down. Remember, though, that in business it is important to agree things in writing. You write down the problem specification so that both you and the client are clear as to what you are going to do.

The problem specification on its own is not enough. If that was all you had then you could end up solving the problem to your satisfaction, only to find that the client did not believe that your solution did the job properly.

To get round this difficulty, you must also agree with the client what the **output** from your solution will be. Again, this might be very obvious. The outputs to some simple problems are given in the table below:

Problem	Output requirements
Produce a set of address labels for customers living in postcode area NE4.	A set of address labels, correctly addressed to the customers in postcode area NE4.
Advertise a disco.	A poster advertising the disco.
Find out which stock needs reordering.	A list of under-stocked items showing the supplier, supplier telephone number and number to reorder.
Find out what the effects of different price increases would be on weekly sales figures.	A computer model that predicts weekly sales figures according to price increases.

You cannot always tell the exact output requirements from the problem specification alone. Information from the client should identify exactly what has to be output. For simple problems, there could be just one output. For more complex problems, there may be a number of outputs.

Once you have agreed the problem and the output requirements, you must identify the **input requirements**. This is the data that will be needed to produce the output. This must be agreed with the client at the start so that you know that what you need will be available. The input requirements for the four problems outlined above are given on page 4.

Problem	Input requirements
Produce a set of address labels for customers living in postcode area NE4.	Customer names and addresses, including the postcode.
Advertise a disco.	Time, place and cost of entry for the disco.
Find out which stock needs reordering.	Stock list with description of the stock, supplier name, supplier telephone number, reorder level, number in stock and number to reorder.
Find out what the effects of different price increases would be on weekly sales figures.	Rules for working out the weekly sales according to prices.

There is one other thing that you have to produce during the analysis – the **performance criteria**. This is a set of statements about the solution that will allow you to decide whether or not it solves the client's problem. Performance criteria are very important and you will learn more about them on page 5. Firstly, however, you need to answer the following questions to ensure that you understand what has been covered so far.

Questions

1 Copy the following sentences and complete them by choosing the correct words from the list below:

analysis *client* *criteria* *evaluation*
five *implementation* *output* *requirements*

There are …….. stages of solving a problem. They are ……., design, …… , testing and evaluation. During the first stage, the problem specification is agreed with the …… . Input …… are determined and so are ……. requirements. Performance ……. will also be agreed so that the success of the solutions can be measured at the end, during ……..

2 Write out the following sentences to show at which of the five stages of the system life cycle each of the described activities would be carried out, when the problem is to produce a poster for a disco:

a Using a desktop publishing program to produce the poster.

b Talking to the organisers to find out details of time and cost.

c Deciding on what font size and colour to use to display the price.

Performance criteria

When the solution is delivered to the client, there must be some way of measuring whether or not it does what is needed. The client and the ICT professional must have some way of agreeing whether or not the solution is a success.

Without any agreed way of measuring success, the ICT professional could have a client claiming that the solution does not do the job properly and refusing to pay for the work. From the other point of view, the client could find that they are being asked to pay for a solution that is not up to standard.

The table below gives some performance criteria that might be agreed for the four problems used as examples on page 3. In each case, the client and the ICT professional could check the solution's performance against the performance criteria to see if it meets them:

Problem	**Performance criteria**
Produce a set of address labels for customers living in postcode area NE4.	The labels must be 7 cm by 4 cm and be printed in a two-column layout, ten to a page. Only labels for customers living in NE4 should be printed.
Advertise a disco.	The poster must show the correct time, place and ticket price for the disco. All the information must be accurate. The poster must be eye-catching, on A4 paper and use colour. All information should be readable at a distance of 5 m by someone with normal vision.
Find out which stock needs reordering.	The list must show only stock items that need reordering. All the items from one supplier must be grouped together. The list must be in columns with the headings: Stock item, Number to order, Supplier, Telephone number.
Find out what the effects of different price increases would be on weekly sales figures.	It must be easy to input the price increase. The resulting change in weekly sales must be shown in an obvious way. The model must predict changes accurately.

Performance criteria are used at the end of a project to decide whether or not it has been a success. However, they are also used in the design stage.

When the solution is being planned, the designer will keep the performance criteria carefully in mind. Consider the poster example from the table above. When the designer is deciding on the layout of the poster, the performance criteria restrict the size of paper that can be used to A4. The designer may want to use a different size but the client specified A4 and, if the final solution is not on A4 paper, the client would be justified in refusing to accept it and, more to the point, refusing to pay for the work.

The designer also decides how to test the solution during the design stage and they will design some of the tests to show that the solution meets the various performance criteria.

Doing the analysis

In your practical work, you will not be able to discuss requirements with a real end-user. Instead, you will be given copies of letters and other documents to examine. You will have to look at these to analyse the problem.

Begin by reading whatever material you have been given. As you read, try to work out what needs to be done; mostly this will be obvious. Once you are sure you know what the problem is and you have written down the problem specification, look for the output requirements (ie what has to be produced, printed or perhaps displayed onscreen).

When you have the output requirements, identify the data or other inputs that you will need to produce the output. Finally, go through all the documents again looking for performance criteria. Be careful when you are writing these. You might think of something really good that the client should have. **Do not put it in!** The performance criteria are what the client thinks the solution should do. It is not up to you to put additional items in. Do make sure, though, that you include everything the client does want.

Practical 1: The Junior Library

Newington Library has a junior section which caters for young people up to the age of 16. Mr Jones, the librarian, is keen to computerise the library's records and has sent a series of memos outlining the requirements of the solution that is required. The first memo from Mr Jones is shown below. He outlines some details about two files that the database will have: members and books.

Memo

Members and books

Obviously, since we are a library, we need to store details about our members and the books that we have.

Members sometimes ask if we have a particular book. Occasionally, they will know the title and the author but, more often, they only remember the title. Some members can only remember one or two words from the title. Often, if a member has enjoyed a book by a particular author, they will ask if we have any more books by the same author.

We only cater for members from the ages of 12 to 16. When a member reaches the age of 17, we write to them, pointing out that they are no longer eligible for membership and offering to enrol them in the main library. We like to maintain a friendly atmosphere in the library and so, when we write to members, we always use their first name at the start of the letter. Of course, the address label has to have the member's full name.

The library service is paid for by the local council and we have to provide them with statistical information from time to time. For example, we must tell them the average age of our members and also the percentage of boys and girls who are members.

Members, books, data and CSV files

The members

1 What does Mr Jones say in the memo that explains why the members file will need to store the following data?

   a Member's date of birth.

   b Member's first name.

   c Member's street, town and postcode.

   d Member's sex.

2 Each member is given a unique six-digit membership number. Why is this necessary?

3 What two pieces of book data must the library store in the books file to meet the requirements outlined in the memo?

4 The table below is part of the design for the members file. Copy and complete the table by selecting appropriate ranges for each data type from the list below. Some ranges may be used twice:

25 characters *20 characters* *15 characters*
8 characters *6 characters* *1 character*

Field name	Data type	Range
Membership number	Text	
First name	Text	
Surname	Text	
Sex	Text	
Date of birth	Date	
Street	Text	
Town	Text	
Postcode	Text	

The books

At present, the longest book title in the library is *A Connecticut Yankee In King Arthur's Court,* which was written by Mark Twain. The author with the longest name is Mary Wollstonecraft Shelley, who wrote a book called *Frankenstein*.

Each book is identified by a unique 15-character book number. For example, *The Lair of the White Worm* by Bram Stoker has a book number 2-47049-70491-X. The final character of the book number is a modulo-11 check digit.

1 Copy and complete the design table below to show the design for the books file:

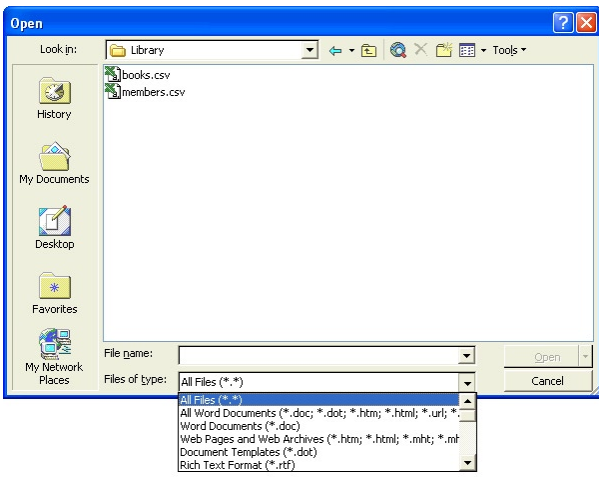
Field name	Data type	Range
Book number		
Title		
Author		

2 The check digit at the end of the book number is used to validate the number when it is entered. A check digit will spot two common errors. What are they?

The data and CSV files

The library has 60 junior members and over 150 books. To save time, the librarian has typed out all the data about the members and the books. This data is stored in two special text files called members.csv and books.csv. Copy these files to your directory. Once you have copied the files to your directory, load Microsoft® Word. Then open members.csv using Microsoft® Word. Normally, Word looks only at files that end with the extension .doc.

You will have to tell Word that you want to open a file with a different extension. The simplest way to do this is to tell Word to look at all files, no matter what the extension is. In Word, go to File/Open to see the first dialog box, as shown on the right. Now click the Files of type drop-down menu. Change the entry from Word Documents (*.doc) to All Files (*.*). You should now see the two CSV files, members.csv and books.csv.



Open members.csv. You will see the members data needed for the library solution. Text data in this format can be imported directly into a database file.

1 Examine the text stored in members.csv and then use the words in the following list to copy and complete the sentences below:

comma *CSV* *database* *field names* *fields* *line* *text*

The file members.csv is a ..... file, which is set out in a special format so that the data it holds can be imported directly into a ..... . A database file stores records and each record is made up of a number of ..... . In the text file, one record takes up one ..... . The data for one field is separated from the next by a ..... . The use of commas to separate data values gives this type of text file its name. It is called a comma separated values or ..... file. The first line of the members.csv file does not contain data. It stores the ..... .

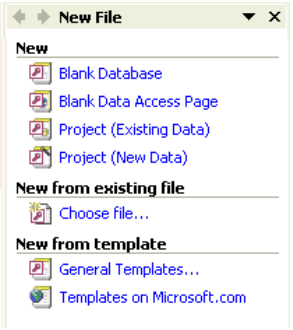
2 Now use Microsoft® Word to look at the other text file, books.csv. Answer the following questions:

a The first line of books.csv is the field names for the data. What three field names are given on this line?

b What is the book number for the third record in this file?

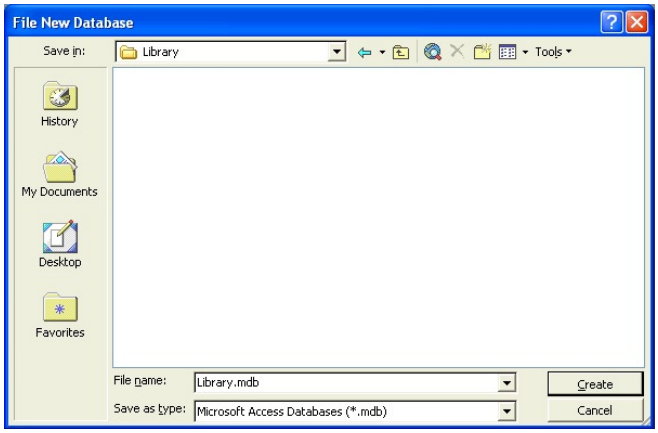
Creating a new database file

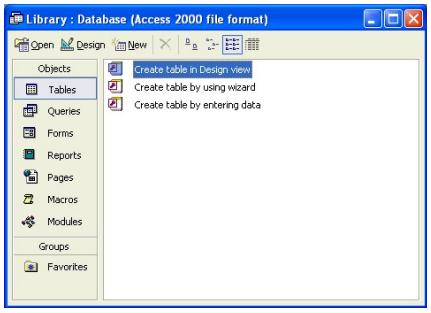
The first step in the solution is to create a database file that will eventually store the data from the CSV files:



1. Load the database program, Microsoft® Access. If the Getting Started panel appears, select Create a new file to open the New File pane. On the New File pane, select Blank database, to create a new database.

2. Enter the file name 'Library' in the File name box. Make sure that the Save in box is your network directory. Then click the Create button to create your blank database.





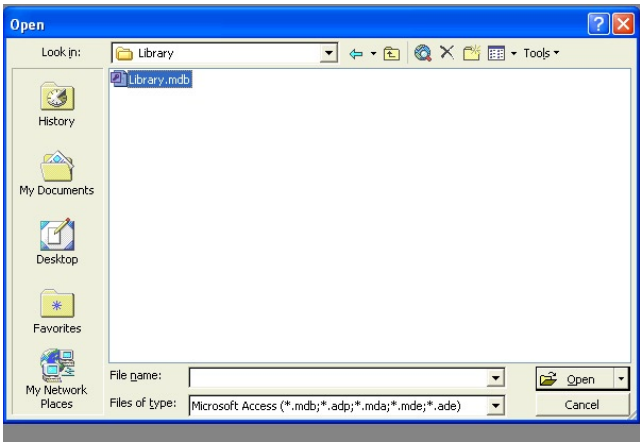
3. You should now see your blank database. There may be various wizards available, but there will be no entries under the table, query or other tabs since, at present, your database is empty. You can now close Access.

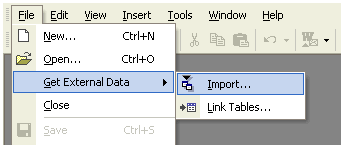
Importing data from CSV files

In this section, you will be taken through the steps needed to import the data from members.csv into your new database file. You will then be expected to import the books.csv data without any further help.

1. Open Microsoft® Access. On the Getting Started panel, click More….

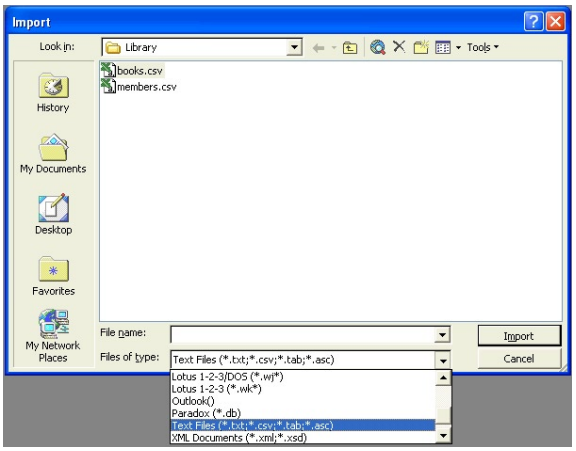
2 Find the blank database file that you created and saved in the previous section of work. Select the database and open it.

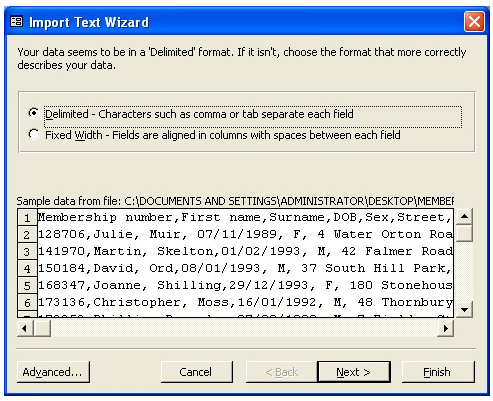




3 Select File/Get External Data/Import.This tells Microsoft® Access that you want to import data from an external file.

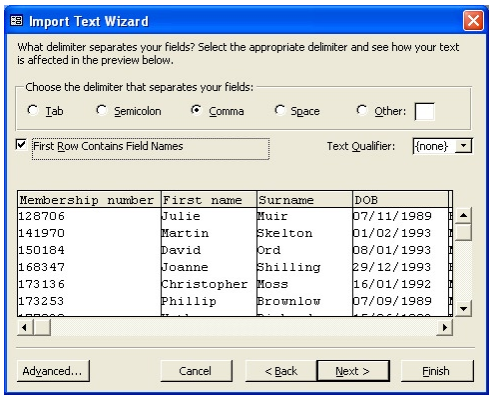
4 Next, you must tell Access what type of file you want to import and where it is. Select Text Files (which includes CSV files) in the Files of type box. Make sure the Look in box is showing the directory that you copied the two data files to. Select members.csv and click the Import button.





5 The Import Text wizard will open the file. There are two different formats that are often used. One is delimited, where a special character – a comma in our case – is used to separate data values. The other is fixed width, where data values are padded out with spaces so that they are the same number of characters. The wizard should detect that the file is delimited, but make sure that this file type is selected before continuing. Then click the Next button.

6 The Import Text wizard will detect that the delimiting character that separates the field values is a comma. However, it will not know that the first row of the file contains the field names. You must click the First Row Contains Field Names box before continuing.



7 When you click the First Row Contains Field Names box, the wizard will grey out the first row to show that it is not data. Check that the wizard looks like the picture on the left and then click the Next button to move on.

8 The wizard will now ask where the data is to be put when it is imported. In Microsoft® Access, data is stored in tables. Access will set up the new table for you and import the data to it. However, you do need to check that Access has correctly identified the data types for each field. You will need your design for the members file, as required by question 4 (see page 8). Make sure that the option In a New Table is selected and then click Next.

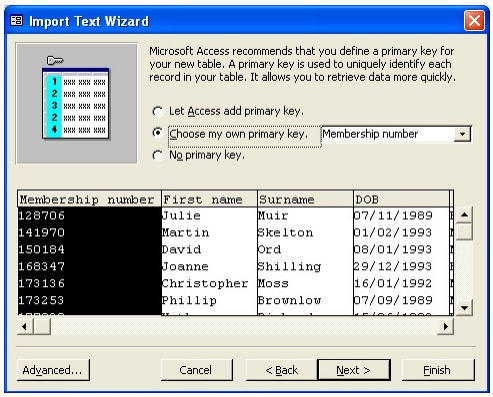
9 The wizard will think that the first field – Membership number – is a number (Long Integer). Your design should show that it is text. Tell the wizard that this field is text by selecting Text from the drop-down list of data types.

You select a field by clicking on the field name at the top of the column. Check that the Date of birth field is set to Date/Time and that all the other fields are set to Text.

When you have checked all the data types, click the Next button.

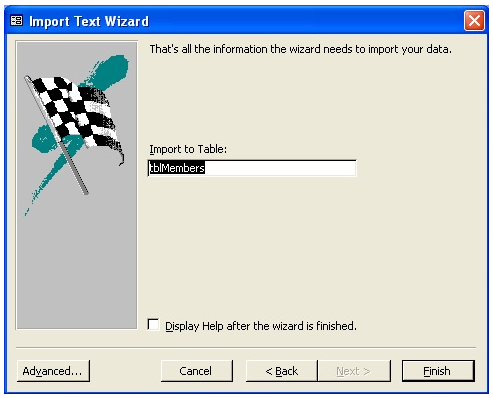
* File corrupted! Call support on 01223 350555 *

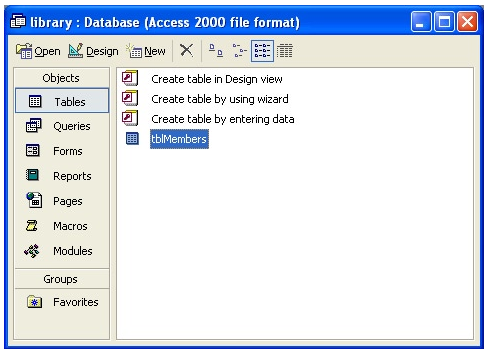
10 Access will offer to add a primary key (key field) to the data. You do not want this to happen. Select the Choose my own primary key button.



11 When you select the Choose my own primary key option, the extra field that the wizard put in will disappear. Make sure that Membership number is selected as the primary key and then click Next.

12 Finally, the wizard will check that the table name it has chosen is suitable. We are going to put the letters 'tbl' in front of the names of tables in the database, so change the Import to Table box so that it contains the word 'tblMembers'. Now click the Finish button.





13 You should now see the database window with the new table, tblMembers. If you open tblMembers, you should see all the member data that has been imported.

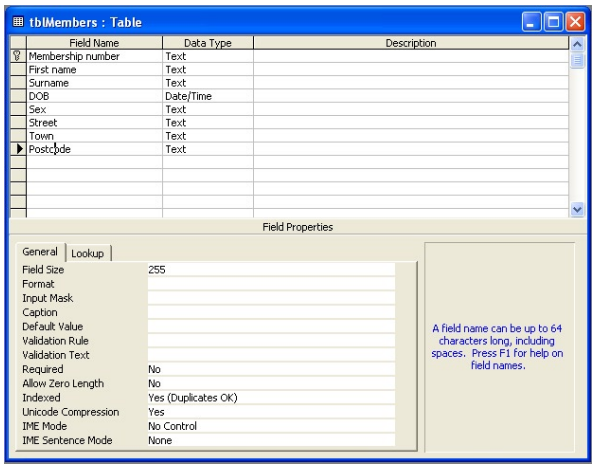
The books data

You have just seen how to import data from a CSV file. Now use the same method to import the data from the books.csv file into a table called tblBooks with the Book number as its primary key. All the fields are of Text data type.

Completing the table

At this stage, you have imported the members and the books data into tables in the database. If you have followed the instructions carefully then each of the fields should have the correct field names (from the first row of data in the CSV file) and the correct data type, which you checked during the import process. However, the range for each field has not been set.

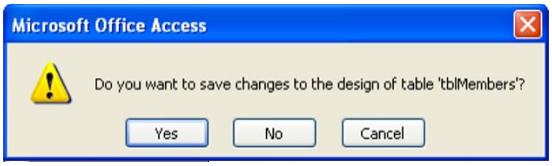
You will need to look at the two designs (see page 8) you completed where you decided what range values to use for the fields in the two files. You are going to open the database and change the structure of the two tables to match your design.



1. Open up your database and make sure that you have the Tables tab selected in the database window. You should be able to see the two tables that you have imported, ie tblMembers and tblBooks. Select tblMembers and then click the Design button to see the table in design view. It should look like the picture on the left. You should see a list of field names and data types. Notice that Membership number has a small key icon beside it showing that it is the primary key field for this table.

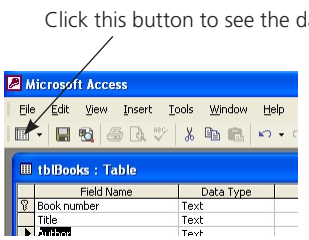
At the bottom left-hand corner of the screen, you will see some additional information about the currently selected field. The field size for text fields will probably be the default value of 255. Select each field in turn and change the field size value to whatever you decided it should be in your design (see page 8).

2. When you have completed setting up tblMembers, close the window. You will be asked if you want to save the changes that you made. Click Yes.

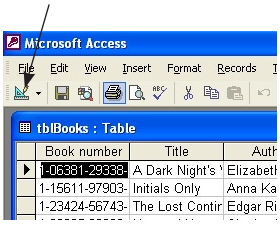


3. Microsoft® Access will then warn you that, if you have chosen too short a length for the text fields, data will be lost. Click Yes.

4. After setting up tblMembers, do the same for tblBooks. You have been working on the tables in design view. You can switch between design and data view using the tool bar. Open up one of the tables in design view and click the Data Sheet View tool to see the data in the table.



Click this button to return to design view

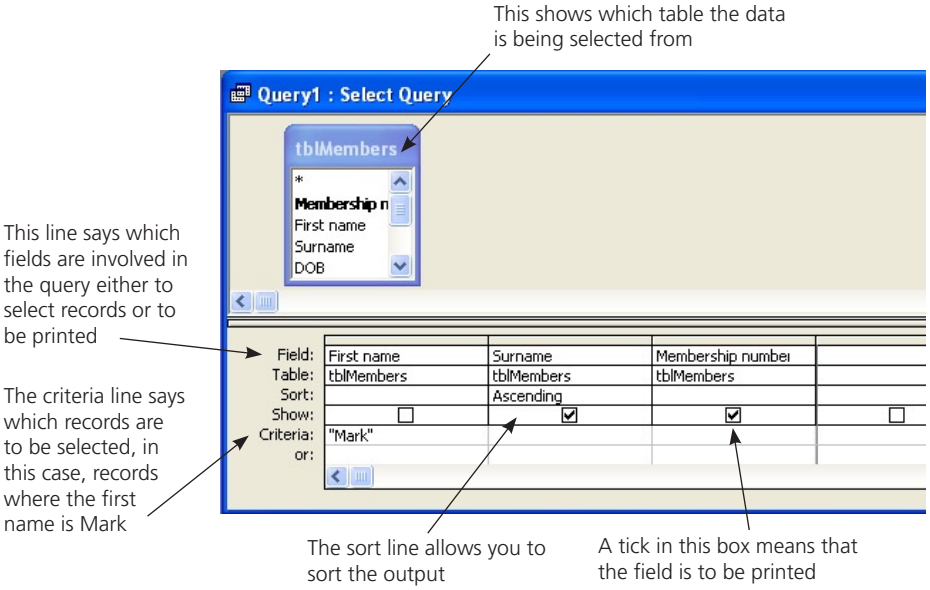


5 In data sheet view you see the data that is in the table. The data sheet view tool changes to a design view tool and the icon changes to a set square, ruler and pencil. Clicking this tool returns you to design view. Practise moving between design and data view.

Selecting the data

So far you have set up two tables to store data. The table tblMembers stores all the data about the members and tblBooks stores all the data about the books. Data view can be used to see all the records and fields in a table. However, normally you would only need to see some fields and some records. Hence, you need a way of telling Microsoft® Access which fields and which records you want to see. This is done using a **query**.

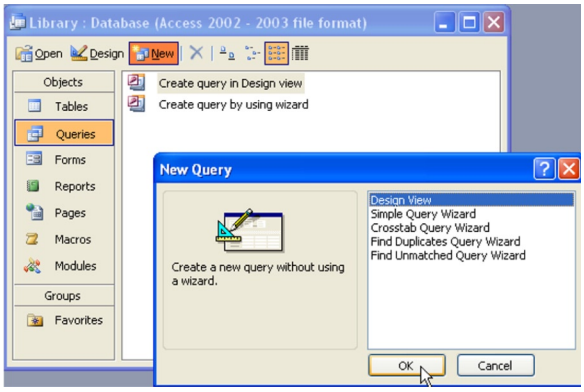
A Microsoft® Access query allows you to select the fields and records that you want to see. Set up a query in design view and then switch to data view to see the information you want.



The sample query above selects all members from tblMembers whose first name is Mark. It prints their surname and membership number and the output is sorted in alphabetical order by surname. Look at the way it is made up. You need to understand which parts of the query do each of the following, ie tell the query:

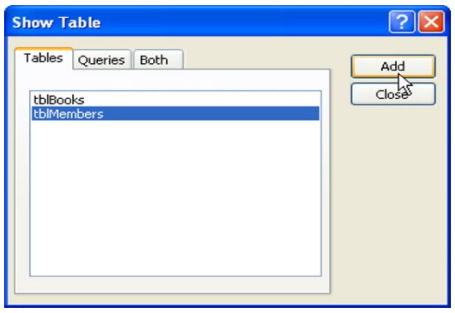
- which table contains the data it is to find
- which fields are to be shown
- how to select the records that are to be shown
- the order in which the information is to be displayed.

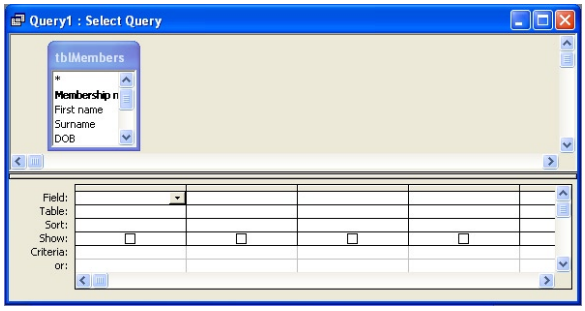
Implementing queries



1. Open the library database and select the Queries tab in the database window. Click the New button and make sure that Design View is selected in the dialog box, then click OK.

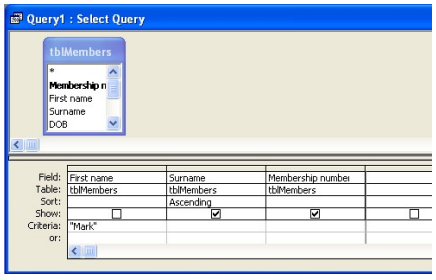
2. Select the table tblMembers in the Show Table dialog box and click the Add button once. Then close the dialog box. You should see tblMembers added to the tables section of the query.

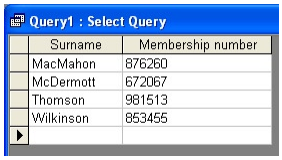




3. The tables section of the query shows the table that the query will select data from. You can see each of the fields that make up the table. If you want to include a particular field in the query, drag it from the table to the query grid below.

4. Complete the query grid by dragging the three fields, ie first name, surname and membership number, to the grid. Type 'Mark' on the Criteria line under First name. You do not need to type the inverted commas, Microsoft® Access will add these to show that it recognises that you have entered text in this space. Clear the tick from the Show line for First name since it is not to be printed. Select Ascending for the Sort line under Surname.



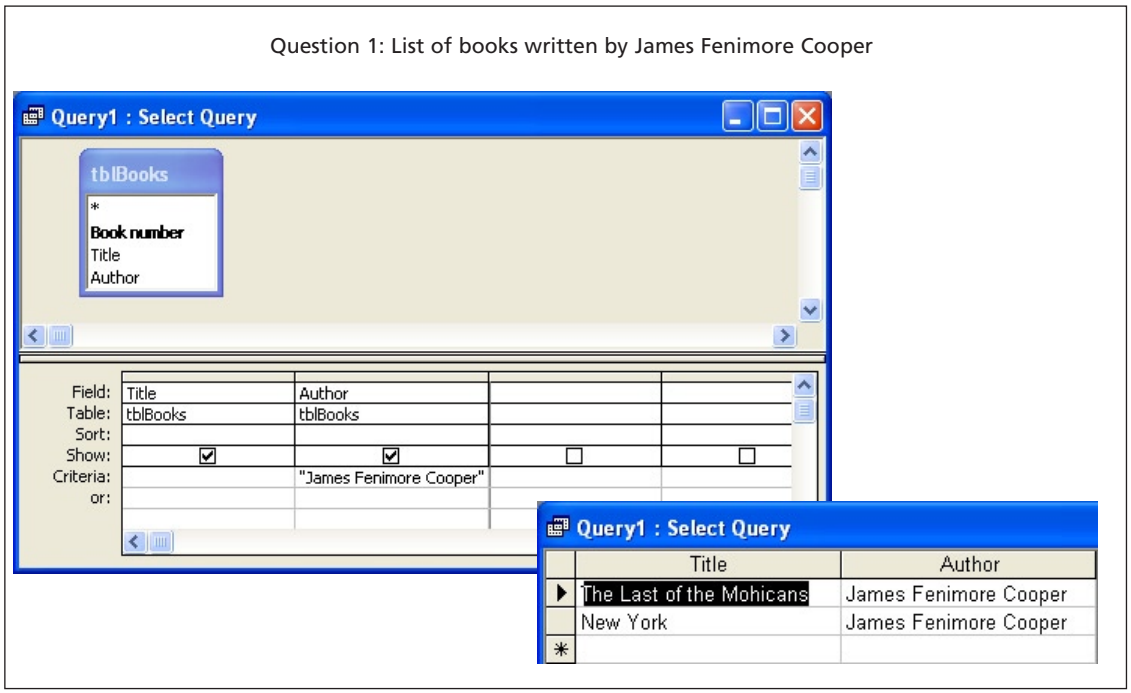


5. Click the View Data Sheet tool (it is in the top left-hand corner) to see the results of the query. You should see four names displayed. Close your query. You do not need to save it.

Queries: Try it yourself

In this section, you are asked to use queries to find information from the database. In each case, you should produce a printout showing the query you used in design view and the results. A sample of what is required is shown below.

You cannot print out the design view of a query directly from Microsoft® Access. A simple way to obtain the design view printout is to use the Print Screen key to copy the computer screen to the clipboard. You can then open Microsoft® Word and paste the clipboard to it. You will then have a copy of the screen showing the query in design view. Print this out. Print the data view directly from Microsoft® Access, cut it out and paste it underneath the design view printout that you have already produced. Write the question that you are answering at the top of the sheet. The final result should look similar to the illustration below:



Complete each of the following tasks.

1 Produce a list of books written by James Fenimore Cooper. Remember to stick the list and the printout of the query onto a sheet of paper as shown in the example above.

2 A member would like you to find out the author and title of a book. They can remember that the first word of the title was 'My' and that the author was female. (Hint: Use the wild card character (ie *) to find titles that start in 'My', ie My*.) Remember to stick your results onto a sheet of paper.

3 A member remembers a book that had the word 'adventure' somewhere in the title. Can you find a list of possible titles? (Hint: This time you need to use the wild card twice in the same criterion.) Stick your list and the printout showing the screen dump of the query onto a sheet of paper.

4 Produce a list of members who were born on or before the 26/6/1990. The list should be sorted in ascending order of surname and it should give the first name, surname and date of birth of the member. Stick your answers onto a sheet of paper as before.

5 Produce a list of members who were born in 1989. The list should give the first name, surname and date of birth and be sorted in ascending order of date of birth. Stick your answers onto a sheet of paper as before.

Adding interaction to queries

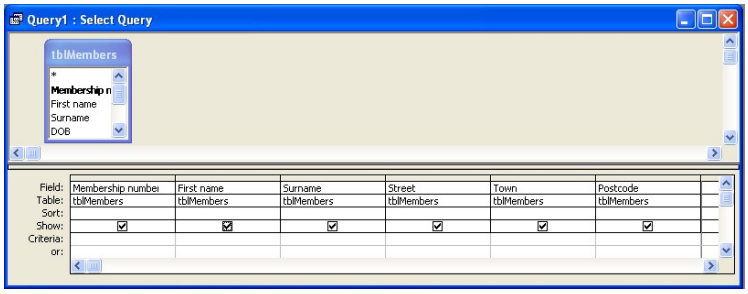
The next step is to make the queries more generally useful. At present, the user would have to create a new query to obtain each separate piece of information. What we need is a way of setting up a general query, so that the user does not need to go into query design mode to find out information.

This is easy to do and an example should make it clear. In the previous section, you entered criteria to tell Microsoft® Access which records to select. Instead of entering the criterion in design view, you can type in a prompt to ask the user to enter the criterion. The prompt is entered in square brackets. An example might be:

[Enter the member's surname]

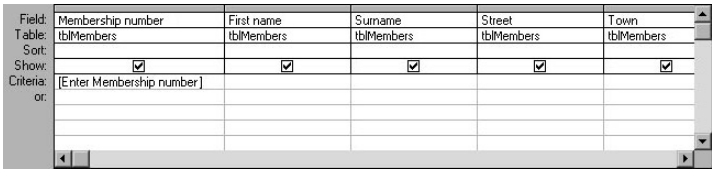
Now, when the query is run, Access will display the prompt and the user can enter the criterion without having to go into query design view. The same query could then be used over and over again.

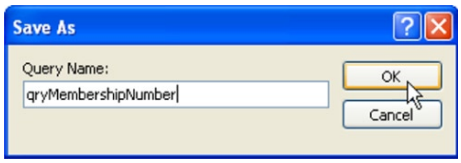
Follow the steps below to see how this works.



1 Create a new query based on tblMembers. Add the fields membership number, first name, surname, street, town and postcode to the query grid.

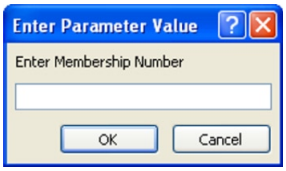
2 Now type [Enter Membership number] in the criteria box for the membership number field. Remember the square brackets; these tell Access that the actual criterion will be entered when the query is run.

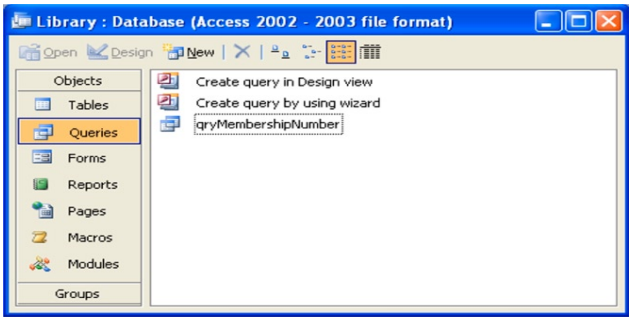




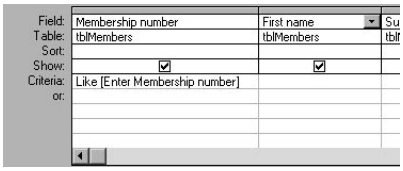
3 Use the File/Save command to save the query. Give it the name 'qryMembership Number'. Notice that we put 'qry' at the start of query names. This makes it easy to distinguish them from tables, which had names starting with tbl. When you have saved your query, close it.

4 Now you can see your query saved in the database window with the Queries tab selected. When you open your query, you will be prompted to input a membership number.





Enter 244046 as the membership number to check that you get Paul Brown's details.



5 The query that you have created is called a parameter query because the criteria row value is put in when the query runs. You may spot a problem with your query. Wild cards do not work. Try it by entering '2*' as the membership number that you want. This should produce a list of members with a membership number that starts with a 2. You will find that you do not get any results with your query. The solution is quite simple. Go back to design view and change the entry on the criteria row, putting the word 'Like' in front of your prompt.

Test your final version of the query as follows:

1 Produce a list of members with a membership number that starts with 2.

2 Produce a list of members with a membership number that ends with 4.

Practical 2: Finding a Book

You are going to use a parameter query to produce an easy way for the librarian to find the full details of a book when just the title or part of the title is known. You will also learn how to design a test plan for a query and to perform testing.

Finding a book analysis

Junior library members often arrive at the librarian's desk asking if a particular book is available. They often know the title of the book but sometimes they know only part of it. The librarian needs a system which will allow her to find full details of books, given part or all of the title.

The performance criteria are:

1 The system must produce full details of the book, ie book number, title and author.

2 It must be possible to find the details when the full title or any part of the title is known.

3 If there is more than one book in the list, the list should be sorted in order of author and then in order of title.

4 The list headings must be in the order of book number, title and author.

5 The solution must be easy to use since the librarian cannot be expected to design queries.

Finding a book design and test plan

You should design your query based on the grid below. However, you will need a little more information about sorting before you can do this. Microsoft® Access sorts the first field in the query grid that has sorting set and then the next. If you want the list sorted in order of author and then in order of title, you need the author field to be further left than the title field.

Unfortunately, the performance criterion for printing asks for the fields to be the other way round. You can get round this problem by putting author and title on the query but clearing the tick so they do not print out. Set the sorting on these two fields. Then put book number, title and author on the grid with the show box ticked so that these fields print. This means that title and author appear on the query grid twice – once to produce the required sorting and once for printing.

Copy the following grid and use it to design your query:

Query name:				Based on table:								
Field												
Sort												
Show												
Criteria												

Test plan

Test plans are an important part of design. The test plan is the section where you say how you will make sure that your solution works. One purpose of testing is to make sure that your solution provides the correct answer. Another is to show that your solution achieves the performance criteria.

Your test plan will, therefore, indicate how you intend to show that your solution works correctly and how it achieves the performance criteria. You do this by saying what data you will input and what output you would expect to get if your solution was working correctly.

It is not enough to show that your solution works under normal conditions. Your testing should try to expose any weaknesses. Therefore, test extreme conditions as well as ordinary conditions. In some cases, you would need to test that your solution dealt with incorrect data as well.

In the case of the particular query that you are designing here, you should select test data that will result in the following outputs:

- A list containing one book only. This is an extreme condition. This output can also be used to show that your solution achieves performance criteria 1 and 4.
- A list containing no books at all. This is an extreme condition.
- A list containing books by several authors with at least one of the authors having several books in the list. This will test that the list is sorted properly, ie performance criterion 3.

To show that your solution fully achieves performance criterion 2, your test data should include a search when:

- the full title is known
- the start of the title is known
- the end of the title is known
- words from the middle of the title are known.

Use the layout on page 22 to complete a test plan for your solution by copying the table and filling in the spaces. The test plan should be done without using the database. If you set up a query, produce some output and then write down what you got as your expected output then you have not really tested your solution at all. Use the printout of the books data to find out what output you would expect to obtain for each of your tests. Your teacher may give you a printed list of books. If not, print out your own version of the list by using the file booklist.doc.

Finding a book implementation

Completing the test plan is the final stage of the design process. Now implement your design. Set up the query exactly as you designed it. Make sure that you give it the name specified in your design.

Show that you have implemented the query by printing out a screen dump showing the query in design view. Stick your printout onto a separate piece of paper and write on it how it is designed to work.

Test	Data used/action	Purpose	Expected output
1	Enter the title: Tom Sawyer	To test the extreme condition when only one book is in the list. Also check that all the required information is present in the correct order.	9-15526-91021-2 Tom Sawyer Mark Twain
2		To test the extreme condition when there are no books in the list.	
3	Enter the title: An*		3-80501-62490-X Anthem Ayn Rand 8-01882-70737-3 Anne of Avonlea Lucy Maud Montgomery 3-80264-08357-9 Anne of Green Gables Lucy Maud Montgomery 5-63759-33390-2 Anne of the Island Lucy Maud Montgomery 8-70613-00341-4 Anne's House of Dreams Lucy Maud Montgomery
4	Enter the title: *red*		
5			4-96950-37861-7 From London to Land's End Daniel Defoe 9-69721-64410-4 Notes from the Underground Fyodor Dostoevsky 5-63759-33390-2 Anne of the Island Lucy Maud Montgomery

Finding a book testing

Once you have the query set up, you must test it by following your test plan. Produce a series of five printouts. Each printout should show one of the tests being done. The printout should have the test number so that it can be cross-referenced with the original test plan. It should also show the information that the query produced and it should have written on it whether or not the test shows the query working.

Finding a book evaluation

Evaluate your solution to the problem. Refer to the performance criteria and to anything in the testing that could be considered evidence for having met any of the criteria.

Practical 3: Mailmerge Letter

When solving The Disco Ticket problem (see Practical 1, *GCSE ICT Practical Course Pack 1*), the mailmerge feature of Microsoft® Word was used. One ticket document was created and merged with a file of numbers. When the mailmerge was carried out, Word created a separate ticket corresponding to each of the numbers in the numbers file.

A Word document can also be merged with data from a database. When the merge is performed, you get a copy of the document for each of the records in the database.

Once the document and the database are linked, fields can be inserted into the document. When the merge is done, each document has one set of data inserted, corresponding to one record. This allows personalised letters to be produced from one basic mailmerge document.

The document can be linked to a query in the database. This allows you to produce mailmerge documents for selected records only. You could, for example, write a letter linked to those members who were born before a certain date.

Memo

Members who are too old

We are all very excited about the new database. It is working very well and we are finding it a lot easier to answer members' questions about book titles and authors.

You will remember that we need to write to members when they reach the age of 17. In fact, to make things simpler, we only write once a year in August. Any member who is 17 on or before the 1 August that year is sent a letter.

Could you design a system that we can use to produce these letters each year?

Obviously, it is important that the system produces letters only for the correct people, so I would like you to test that your solution works using the members' data I supplied. That data was correct for the year 2006 so you should be able to produce letters for members who were 17 on or before 1 August 2006.

I have provided a Microsoft® Word document that is a copy of the letter we sent out last year and I would like your solution to use the same layout.

Mailmerge design and test plan

Read the memo above, which was written by the librarian Mr Jones, about members who are too old for the junior library.

Choice of software

Your solution to this problem will use Microsoft® Word and Microsoft® Access. Explain what part each of these two applications will play in your overall solution and say what features it has that make it suitable.

a What will Word be used for in the solution?

b Why is Word suitable for this task?

c What will Access be used for in the solution?

d Why is Access suitable for this task?

Performance criteria

Find two performance requirements that Mr Jones identifies in the memo. Write them out on a separate piece of paper.

Design of the letter

Since you have already practised designing documents in earlier practicals (see *GCSE ICT Practical Course Pack 1*), you are not going to design the letter. However, you will need to do some preliminary investigation so that you understand how the letter will have to be produced when the solution is implemented.

Last year's letter is stored in the directory with the other files that you have used for this problem. It is called Mletter.doc. Copy it to your own directory.

Now print out a copy of the letter. When you create your mailmerge document, some of the information in the letter will be entered as merge fields. There are seven different pieces of information on the letter that will be inserted as merge fields when you create the letter. One piece of information appears twice.

Circle and label each piece of information that will be a merge field in your solution. For example, you will circle the word 'James' and label it 'First name'. Make sure that you use the correct field names from the database.

Now complete the design for the top part of the letter by copying the diagram below and writing the field names in the appropriate spaces. Each field name is placed between triangular brackets to show that it is a merge field rather than ordinary text. Use the printout of the letter and your labels to help you complete this design.

<
> <
>
<
>
<
>
<
>

August 2006

Membership number: <
>

Dear <
>

Our records show your date of birth as <
>, which means that you were 17 years old on or before 1 August this year. You are, therefore, no

Design of the query

Copy and complete the grid on page 25 to show the design for the query that will allow the librarian to select records of members who are 17 years of age on or before 1 August 2006. You may be tempted to use a parameter query for this task, so that it can be easily used next year. However, this would cause some problems when you link the query to the letter. You may need some extra space to write out your criterion.

Name of query:				Table that query is based on:			
Field							
Sort							
Show							
Criteria							

Design of the test plan

Read the following test plan carefully and then copy and complete it:

I plan to test my mailmerge solution using the data for 2006. Firstly, I will use a query to sort the members by date of birth and copy out the names and dates of birth of any members born in 1989. Then I will look at the list I have copied and circle the names of any members who are 17 years of age on or before 1 August 2006. This will tell me which members should receive a letter.

When I have implemented my solution, I will perform the mailmerge. I will compare the letters produced with my expected output to see if they are the same. If they are, I can assume my solution is working correctly.

The members born in 1989, in ascending order of date of birth are [you produced this list as part of your answer to the questions on page 18]:

Member's name	**Date of birth**

Ensure you have circled the members who should receive a letter.

Mailmerge implementation

There are three stages to implementing your solution:

1 Set up the query in the database.

2 Create the basic mailmerge document, linking it to the query that you have already set up.

3 Perform the mailmerge to print the letters to the members.

At one point or another in this course you have already done each of these three things, so this time there is no step-by-step guide to help you (see Practical 1, *GCSE ICT Practical Course Pack 1* and Practical 1, page 16).

Set up the query. While still in the database, test that your query works. It should produce a list of two members who were 17 years of age on or before 1 August 2006.

Now load in your copy of the letter. Use Tools/Letters and Mailings/Mail Merge wizard… to create a mailmerge document (of type Letters, using the current document). When you open the data source, ensure that you choose files of type Access Database. When you select the database, you will have the opportunity to link your letter to any table or query in the database. Make sure you link to the query that selects the members who were 17 years old on or before 1 August 2006.

Return to the letter and replace each of the various data items with merge fields, following your design plan to create a mailmerge document.

Finally, go to Step 5 of the wizard to preview your letters. There should be two letters – if you have made any mistakes, go back and correct them. If not, proceed to Step 6 and print out your letters.

Mailmerge evaluation

Evaluate your solution against the two performance criteria that you identified. Refer to your test printouts to back up any claims that you make in your evaluation. Complete your evaluation by discussing how easy it will be for the librarians to use this solution in future years. How could the solution be improved?

Practical 4: Library Book Issue

Book issue analysis

Read the following memo about the book issue system:

Memo

Book issue system

Thank you for the work that you have already done. We were wondering if it would be possible to update our book issue system.

We need a system that will allow us to keep track of which member has which book on loan and the date when the book was issued.

I know that many libraries use bar code systems which allow them to record this type of information. I realise that we do not have the equipment to do this.

Would it be possible to set up system where the bar code numbers were entered manually, so that we could evaluate whether or not it would be worthwhile buying bar code readers? The system should only allow valid data to be input.

When books are issued, we would like to have the member's name and the book's author and title automatically displayed on the screen.

If we are using a computer then it would seem reasonable to have the date of the loan input automatically so that the librarian doesn't have to type it in.

If a member has had a book for more than three weeks, we send a postcard to them. The system must automatically produce address labels to stick on these postcards. Of course, we do not want to repeat labels that have been sent out in previous weeks.

1 When a book is issued, three items of data must be input. What are they?

2 There are two distinct sets of output required from the system. These are described in the memo. One set is temporary and the other is printed. What are they?

3 Identify the performance criteria for the book issue system. Copy and complete the following table:

Number	Hint	Performance criteria
1	Screen display.	
2	Validation.	
3	Date of loan.	
4	Address labels.	

4 Since the database already stores the members' and books' data, it is necessary to use Microsoft® Access.

a What other application could be used to help solve this problem?

b What would this application be used for?

Book issue design

When a book is issued, the librarian needs to record the membership number of the member borrowing the book, the book number and the date. No other information is needed. If the membership number is known, the member's details can be found from tblMembers. If the book number is known, the book details can be found from tblBooks.

A new table will be needed to store the loan details. The design for this table is shown below:

Table name: **tblLoans**			
Field name	**Data type**	**Range**	**Comment**
Loan number	Autonumber		Key field
Membership number	Text	6 characters	
Book number	Text	15 characters	
Date	Date		Should automatically be today's date

The Membership number and Book number fields in this table will be used to link to the two other tables that store membership data and book data. It is very important that the data type and ranges chosen for these fields are exactly the same in this table as they were in the original tables, otherwise the links between the tables will not be possible.

When a book is issued, the librarian will input the membership number and the book number using a form displayed on the computer screen.

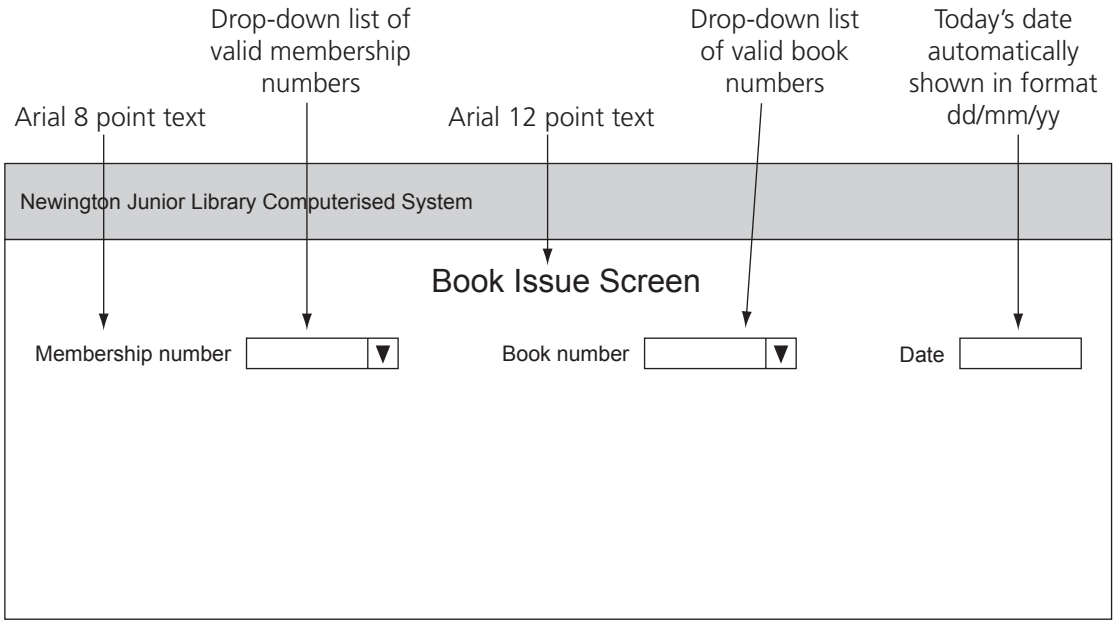
These will be entered using a special drop-down list of valid numbers, which will ensure that the design meets the requirement to allow only valid data to be input. When the entry screen design is produced, the drop-down lists are shown using the symbol on the right.

Drop-down list symbol

When the librarian clicks on the arrow, a list of valid numbers will be shown. The librarian can also type in the number, but only valid numbers will be accepted.

Part of the input screen design is shown on page 29. Copy and complete the design to show where the other information that the librarian wants to see on the input screen will be displayed. The information will be displayed in ordinary boxes, like the date box – not in drop-down lists. Reread the memo on page 27 if you do not remember the additional information that is to be shown on the input form.

The form will be given the name frmBookIssue.



The form used for the input screen will be based on a query. The query is needed to bring the data together from the different tables. The design for the query is given below.

Note that, since the query is using fields from more than one table, we should now say which table each field is from. The membership number and book number must be taken from tblLoans, since we are creating a new record in this table that needs new values for these two fields.

	Query name: qryBookIssue			Based on tables: tblMembers, tblBooks, tblLoans				
Field	Loan number	Membership number	Book number	Date	First name	Surname	Author	Title
Table	tblLoans	tblLoans	tblLoans	tblLoans	tblMembers	tblMembers	tblBooks	tblBooks
Show	✓	✓	✓	✓	✓	✓	✓	✓
Criteria								

A design layout is also needed for the address label that will be stuck onto the postcard that is sent as a reminder to members with overdue books. As you saw in The Disco Ticket problem (see *GCSE ICT Practical Course Pack 1*, page 7), labels come in various standard sizes. A suitable size for an address label to be used on a postcard is about 4 inches wide by 2 inches high.

Complete a design to show how the label will be set out. You need to show the layout for a single label and also for the whole page. (Refer back at the layout diagrams for The Disco Ticket problem if you are not sure what to do.)

The label will be linked to a query in the database called qryOverdue, which automatically selects members who have books that have become overdue within the last seven days. If a book has become overdue, it must have been taken out between three and four weeks ago. Books taken out more than four weeks ago and not returned will already have been dealt with.

The criterion <Date()-21 will select book loans that were made over three weeks ago. Date() is a way of referring to today's date and Date()-21 is the date 21 days (three weeks) ago.

However, you need to exclude books taken out more than four weeks ago, since reminders will already have been sent. The criterion >=Date()-28 will ensure that only loans taken out three weeks ago are included. You could use the criterion <Date()-21 AND >= Date()-28 or, as shown in the query design below, you can use the date field twice in the same query:

	Query name: qryOverdue			Based on tables: tblMembers, tblLoans			
Field	First name	Surname	Street	Town	Postcode	Date	Date
Table	tblMembers	tblMembers	tblMembers	tblMembers	tblMembers	tblLoans	tblLoans
Sort							
Show	✓	✓	✓	✓	✓	✓	✓
Criteria						<Date()-21	>= Date()-28

Book issue test report

The book issue system will be tested by adding six records to make sure that data can be added using the form displayed on the input screen.

Once it has been shown that the system will accept valid data, it will be tested to make sure that invalid data cannot be entered. An invalid membership number and an invalid book number (with the wrong check digit) will be entered. Also, the same book will be issued to two different members.

A further three records will be added on a different day to make sure that the date is being stored automatically. The query will need data from tblLoans to identify the overdue books and from tblMembers to provide the name and address of the member.

The labels for the postcards will be tested by changing the dates in tblLoans, so that of the nine records entered, three are overdue, three are not overdue and three are more than 21 days overdue. In each group of three, one of the records will have a date that is the boundary date for the group.

Data for Test 1: Set of six book loans to be input

The purpose of the test is to ensure that the screen entry form is usable, that it displays the information as required and that data is correctly added to tblLoans.

Input	Action	Expected output/results
141970, 1-59507-33132-7 177028, 3-69387-22900-6 393168, 2-47049-70491-X 946287, 9-78636-58740-0 944377, 1-29388-99850-1 514148, 6-76573-07826-2	Data input using the data entry form frmBookIssue.	The member's name, the book title and author should be displayed automatically on the form as the membership number and book number are entered. Screen display should show: Martin Skelton The Dawn of A Tomorrow, Frances Hodgson Burnett Kathryn Richardson Options, O Henry Rebecca Elliott The Lair of the White Worm, Bram Stoker Gemma Berson The Bridge-Builders, Mark Twain Laura Brown The Man Who Knew Too Much, Gilbert K Chesterton Stephen Reed Uncle Tom's Cabin, Harriet Beecher Stowe Changes to tblLoans: Six records consisting of the membership numbers, book numbers and the current date added to tblLoans.

Data for Test 2: Invalid data entered

The purpose of this test is to check that the system will not accept invalid data.

	Input	Action	Expected output/results
2.1	123456, 1-29388-99850-1	Data entered using data entry form frmBookIssue.	No member details shown Book details: The Man Who Knew Too Much, Gilbert K Chesterton Error message indicating that data is invalid. Data not accepted. Following action (2.2) carried out without closing the form.
2.2	279882, 1-29388-99850-2	Membership number changed to a valid number. Book number changed to an invalid number by putting the wrong check digit on.	Member's name, Adrian Reed, displayed. Book details should blank out. Error message indicating that data is invalid. Form kept open for next set of data.
2.3	279882, 1-29388-99850-1	Data entered.	Screen display now showing: Adrian Reed The Man Who Knew Too Much, Gilbert K Chesterton This book has already been issued to Laura Brown. Error message indicating that the book is not available.

Data for Test 3: Set of three book loans input

The purpose of the test is to ensure that the automatically generated date has changed. The test must be carried out on a different day to test 1:

Input	Action	Expected output/results
Three more records added on a different day. 635342, 8-03750-17864-2 985414, 1-15611-97903-2 941176, 1-28895-89933-3	Data entered using on screen data entry form.	Screen display should show: Lynsey Watson The New Revelation, Arthur Conan Doyle Neil Hall Initials Only, Anna Katharine Green Jubilee Macro Haunted-House, Charles Dickens Changes to tblLoans: Three records added to tblLoans with current date.

Test 4: Dates in tblLoans altered

Nine dates have been altered so that three members have books that are newly overdue this week and need reminders, three have books that have been overdue for more than three weeks and three have books that are not overdue. The actual dates used for this testing will need to be chosen on the day that the testing is done. The table shows how the dates will be chosen. Leave the last column blank for now:

Record to be changed	Date to be chosen for testing	Reason for use	Actual date used
141970 (Martin Skelton)	22 days ago	Book is just overdue.	
177028 (Kathryn Richardson)	25 days ago	Typical data for an overdue book.	
393168 (Rebecca Elliott)	28 days ago	Book is overdue just within this week's reminder.	
946287 (Gemma Berson)	29 days ago	Book is overdue just outside this week's reminder.	
944377 (Laura Brown)	40 days ago	Typical data for overdue but outside this week's reminder.	
514148 (Stephen Reed)	365 days ago	Extreme data.	
635342 (Lynsey Watson)	21 days ago	Book is only just not overdue.	
985414 (Neil Hall)	7 days ago	Typical data for book that is not overdue.	
941176 (Jubilee Macro)	today	Extreme data for a book that is not overdue.	

Input	Action	Expected output/results
Dates in tblLoans changed according to table above.	Labels produced.	Address labels produced for: Martin Skelton Kathryn Richardson Rebecca Elliott The addresses should be those stored in tblMembers for these three members.

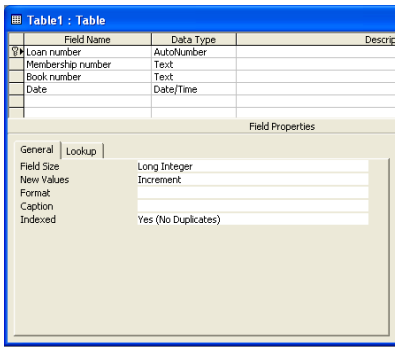
Book issue implementation

The table, query and form needed for the book issue system have now been designed. The section that follows takes you through the implementation.

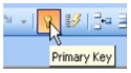
Firstly, set up the table, then create the query that the form is based on, and finally set up the form. You already have experience of doing some of the tasks that are required in this implementation. Where this is the case, you will only be given outline guidance of what to do. If the task is something that has not been covered earlier, more detailed instructions will be provided.

Work through the instructions that follow to set up the book issue system.

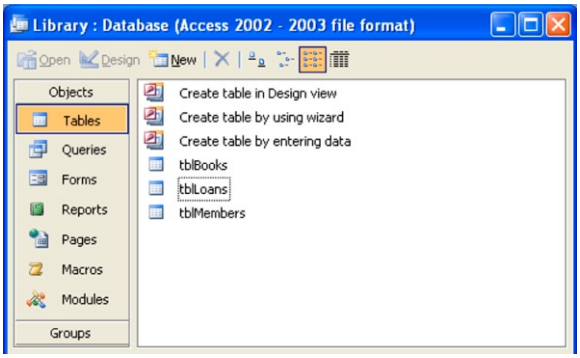
Setting up the table



1. Set up the table tblLoans, according to the design on page 28. The data type counter that you need for the loan number field is available from the drop-down list of data types. You will need to set loan number as the key field. To do this you should place the cursor anywhere on the loan number row and click the primary key tool on the toolbar. This has an icon that looks like a key as shown below.



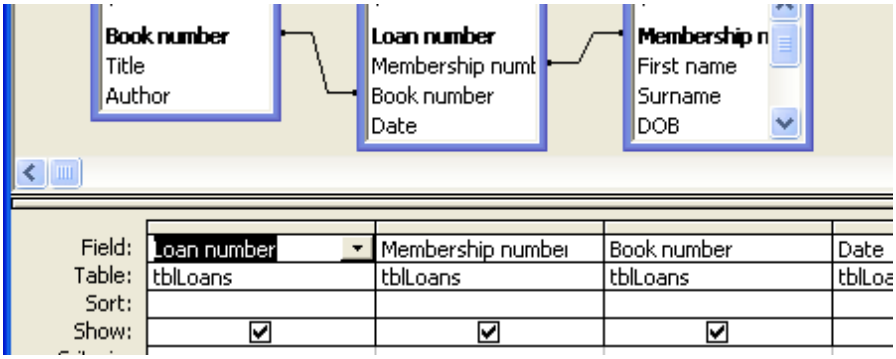
2. Close your new table, saving it with the name chosen in the design – tblLoans. You should now see three tables in the database view window, ie tblBooks, tblLoans and tblMembers.



Setting up the query

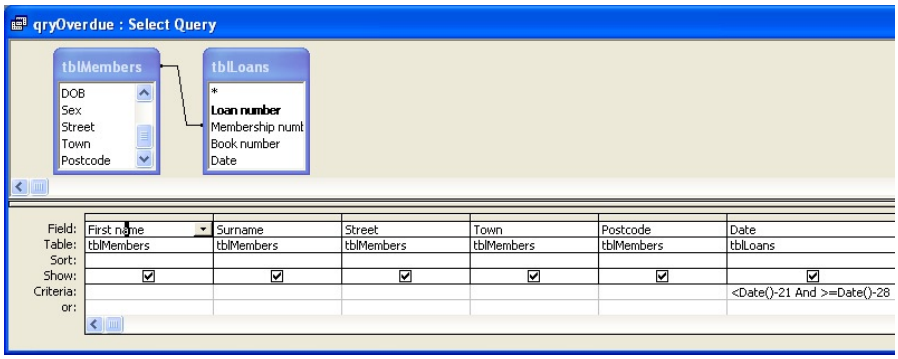
1. Now set up the query qryBookIssue following the design given on page 29. You will need to add all three tables to the query. Microsoft® Access should show the links between the tables with lines, as in the diagram on the left. If these links are not present, click on the appropriate field in either tblBooks or tblMembers and drag to the corresponding field in tblLoans. This will set up the link that Access needs to bring the correct data from the different tables together.

2. Make sure that you drag the membership number and book number fields from tblLoans and not from tblMembers or tblBooks.



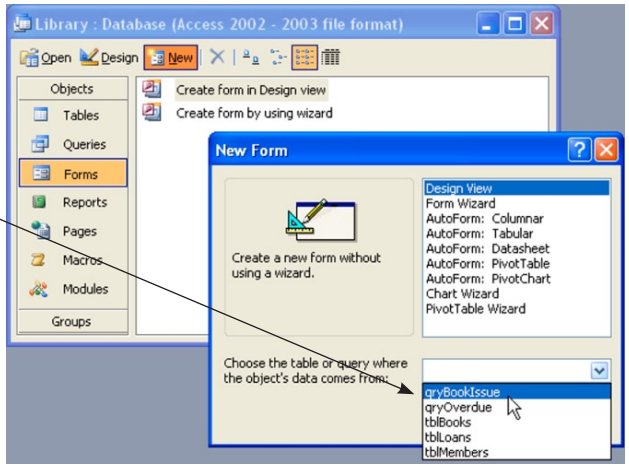
Do not forget the other fields – following your design on page 29. Save your new query as qryBookIssue.

3



Next, set up the other query that the book issue system will use. This is qryOverdue and the design for it is shown on page 30. This query involves two tables, tblMembers and tblLoans. The link between the tables should appear automatically but, if it does not, then drag from membership number in tblMembers to membership number in tblLoans. Save this new query.

Setting up the form

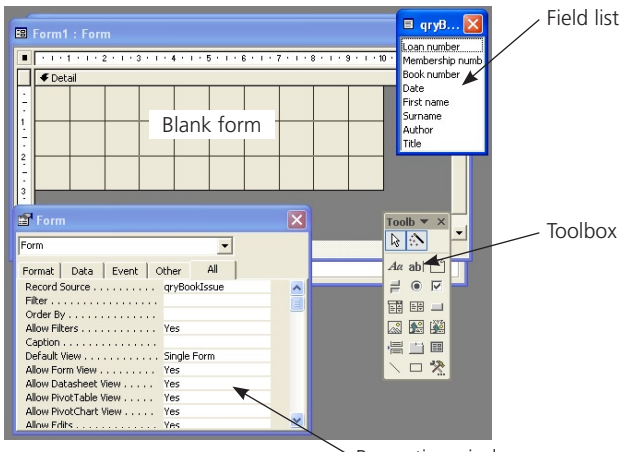


Select qryBookIssue from the drop-down list to base the form on the correct query

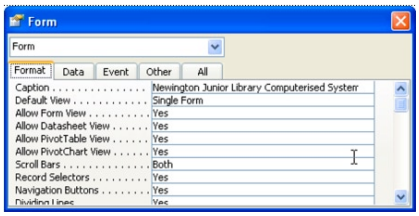
1 Now you need to create the data entry form for the book issue system. In the database view, select the Forms tab and then click New to create a new form. Make sure that Design View is selected and that you have used the drop-down list to tell Microsoft® Access that the form is based on qryBookIssue. Then click OK and you will be presented with a blank form in Design View.

2 Take a few moments to familiarise yourself with the layout of the form design screen.

Use the field list, toolbox and properties buttons on the tool bar to make sure that you can see the field list, the toolbox and the properties window.

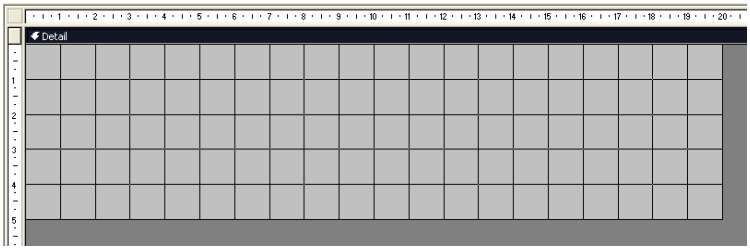


* File corrupted! Call support on 01223 350555 *



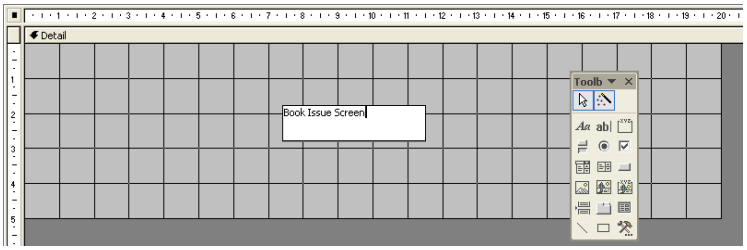
3 Make sure that the properties window has Form at the top. If it has not, use Edit/Select Form to make it so. Enter the Caption text: Newington Junior Library Computerised System.

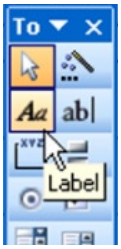
4



Drag the sides of the form out until the form is 20 cm wide and 5 cm high. Microsoft® Access takes its length unit from the value set by the regional settings icon in control panel. (If your computer is set to work in inches then make your form 8 inches by 2 inches.) You may find that 20 cm is too large to fit on your screen. This depends on the screen resolution you are using. If this is the case, adjust the width of the form so that you can just see the whole form on your screen.

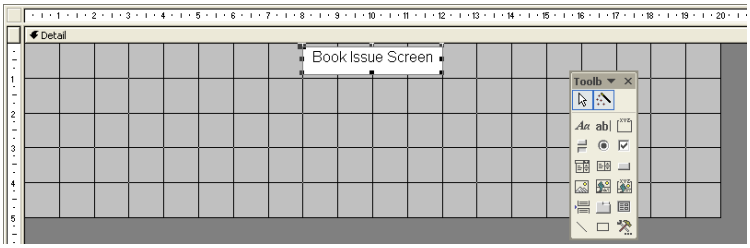
5





Click the Label tool and draw out a label near the centre of the form. Do not worry about the exact size or position at this point. Enter the text Book Issue Screen in the label.

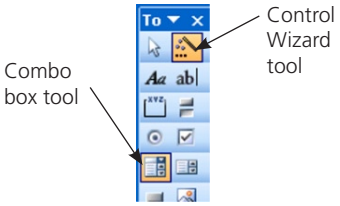
6

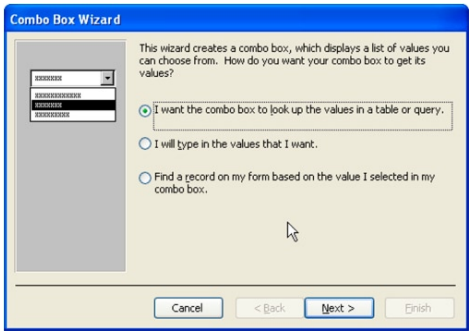


Click anywhere on the form to get out of the label and then click back onto the label so that it is selected. You will know that the label is selected if it has the eight black handles showing. With the label selected, you will be able to set the font to Arial 12 point, centred. You will also be able to position the label at the top of the form as shown. The label in the picture has had its background set to white so that it shows up more clearly. Your label will have a grey background.

7 Before continuing with setting up the form, save what you have done so far. Use File/Save to bring up the Save dialogue box. Save your form as frmBookIssue.

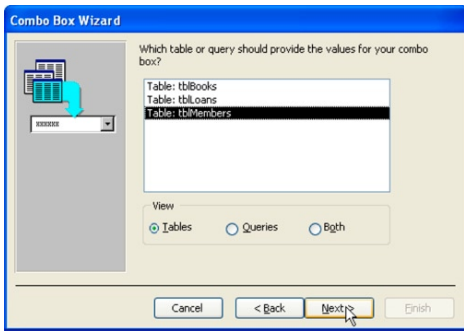
8 Now put a drop-down list of membership numbers on the form. Use a wizard to make this task easier. Make sure that the control wizards are on. To do this, ensure that the control wizards tool on the tool bar is pressed. Then select the combo box tool and draw out a combo box control somewhere near the middle of the form. The wizard should take over straight away – see step 9.

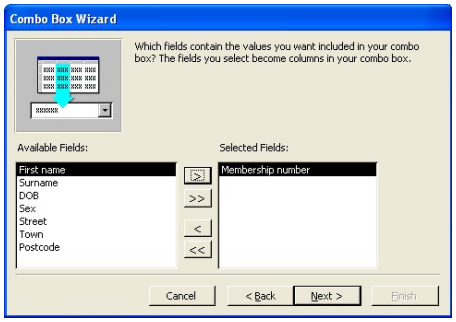




9 Firstly, the wizard asks where the data for the list will come from. Accept the default setting, ie I want the combo box to look up the values in a table or query. Click Next to move onto the next step of the wizard.

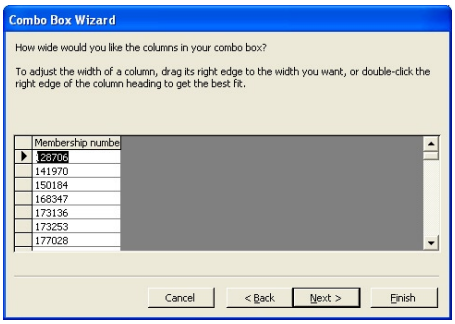
10 The membership numbers are all stored in tblMembers, so select this table as the source of data for the combo box. Click Next to go onto the next step.

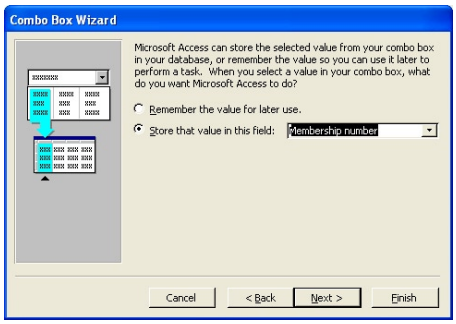




11 You only want to see one piece of data from one field, ie membership number, in your combo box. Make sure this field name is highlighted and then click the single arrow to move it from the Available Fields list on the left to the Selected Fields list on the right. Click Next to go onto the next step.

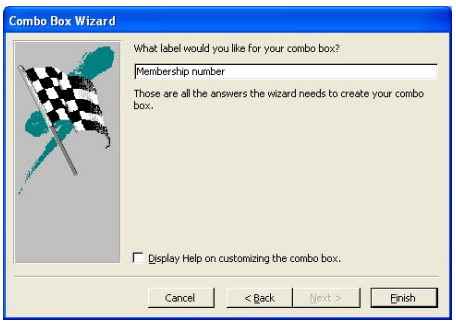
12 Access will show some sample data. This should be a list of membership numbers. If it is not then you have gone wrong somewhere. Click the Back button to go back through the steps to find and correct your mistake. If everything is OK, click Next to go onto the next step.

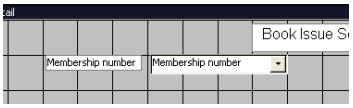




13 Now you need to tell Access where to store the membership number when it is selected in the combo box. Remember that you are entering a new record in tblLoans. You want to store the membership number in the appropriate field of tblLoans. This is why it was important to make sure that, when you set up the query, you dragged Membership number from tblLoans and not from tblMembers. Select the Store that value in this field button and then select membership number from the drop-down list of fields. Click Next to go onto the final step.

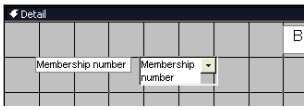
14 Finally, the wizard will ask you what label you want for the combo box. Enter the text 'Membership number' and then click Finish.





15 You will now see your combo box. You will not see the list of membership numbers in design view – you will see that later in form view (see page 38). The label is to the left of the combo box. It is probably too short to show all the text. If you select the label, you can lengthen it by dragging the handles that appear. The label and the combo box can be moved independently of one another by dragging the large handle at the top left-hand corner of each.

16 Position and size the label and the combo box so that they look like the diagram to the right. Select the label and use the properties window to change the font to 8 point Arial. Do the same for the combo box.

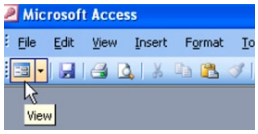


Review your work thus far

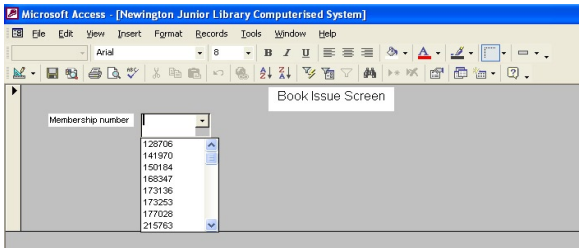
So far you have created a form, added a caption, put a label 'Book Issue Screen' on the form and created a drop-down list of membership numbers. Before carrying on with the implementation, see what the form looks like in use.

You will remember that there are two views for queries, ie design view and data view. There are three views for forms. They are design view (which is the view you have been in so far), form view and datasheet view.

Datasheet view shows the form in table format. You need form view to see the form looking the way that you have designed and set it up. Follow the instructions below to see what your form looks like so far:

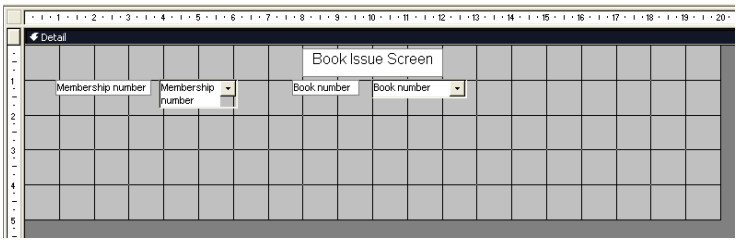


1. The View tool at the top left-hand corner of the Form Design tool bar switches from design view to form view. Click the tool  to see the form in form view.



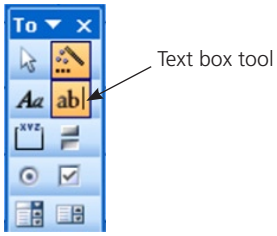
2. You should see your form with the caption you entered along the top edge. If you click the drop-down list, you will see the list of membership numbers. Do not select one or you will get an error message when you close the form. You will also see that the tool that you used to get form view has changed to the Design View icon. Go back to design view. If you get an error message, click OK and try using Edit/ Undo current record before going to design view again.

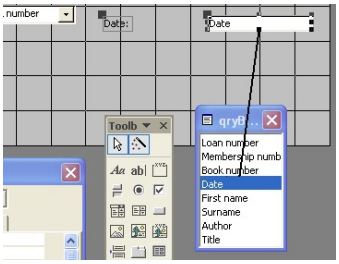
Completing the form implementation

1. 

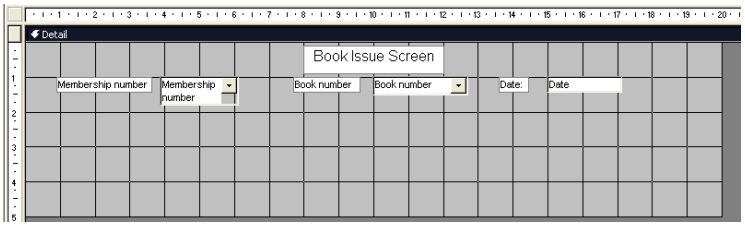
Back in design view, add another combo box that contains all the book numbers. This combo box should look up its data in tblBooks, it should show only one field (ie Book number) and it should store its value in the Book number field of tblLoans. It should have the label 'Book number'. The font for the label and combo box should be Arial 8 point. Your labels will have a transparent background – their colour will look the same as that of the form.

2. The next step is to put a text box on the form. This will be there to enter the date. Select the text box tool from the toolbox and then drag the Date field from the field list onto the form. A text box, linked to the Date field in tblLoans, will appear on the form.



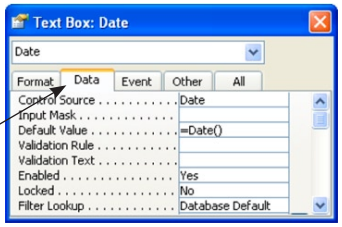


3

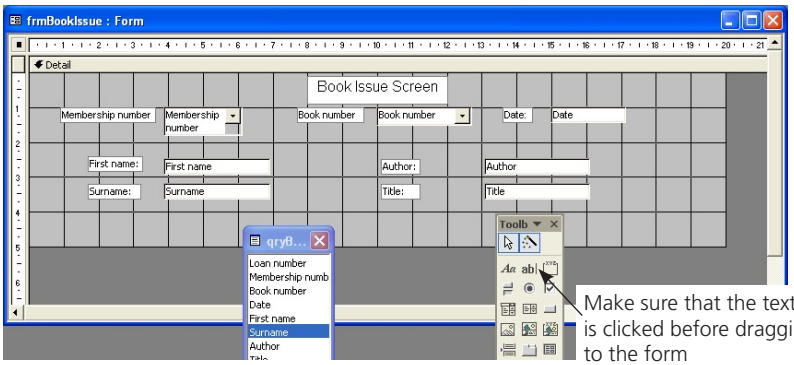


Position the Date text box according to the design and use the properties window to change the font to 8 point Arial.

4 One of the performance criteria for the book issue system was that the date of the loan should be input automatically. You need to set the property for the date text box so that this happens. To do this, you need to use the date function to put today's date as the default value for the text box. Make sure that the properties window is showing and select the date box. Select the Data tag on the text box properties window. Find the row for Default Value and enter the function =Date()

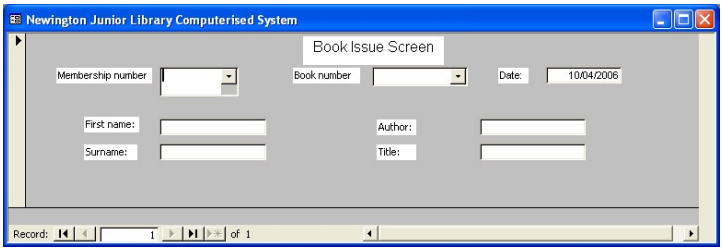


5



Make sure that the text box tool is clicked before dragging fields to the form

The final stage of the implementation is to put the text boxes onto the form to show the additional information that the librarian wants to see. The text boxes will show the member's name and the book's title and author. Your form design as prepared earlier (see page 29) shows where these boxes should be placed. It is important that the text boxes are linked to the fields. Select the text box tool (as you did for the date) and drag the field from the field list to the form. Your layout may differ from the one shown in the diagram. Follow your own design rather than the diagram. Save your completed form.



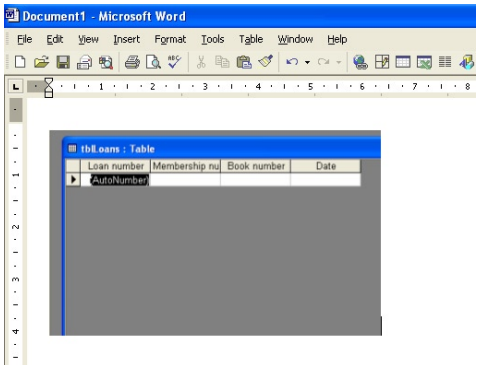
Testing the form

You are now going to perform Test 1 to test that your form works. This testing will be carried out in three stages. The test report on pages 44 to 46 (**Student worksheet 1**) will bring all the test results together. You will add to this test report as each part of the testing is completed.

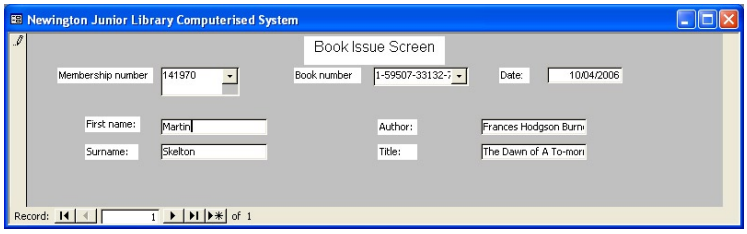
Test 1 should show the addition of six new records to tblLoans. Before you add the records, you should show that the table started out with no records. You can then add the six records and show the table again. This will show that the six records have been added to the table as expected.

Test 1: Following test plan 1 as given on page 30

1. Begin your testing by opening the database table tblLoans in data view. You need to show that, before testing starts, the table contains no records. Copy the screen and paste it into Microsoft® Word, PhotoEditor or any other available program that will let you crop, resize and print pictures.



2. Crop the image so that it shows the empty table, is roughly square and is 2.5 inches wide. Print a copy of the picture and cut out the table. Stick it into Box A of the test report on page 44.

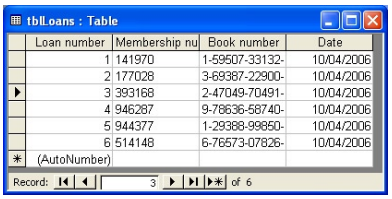
3. 

Close the table and open up your form frmBookIssue. You are carrying out Test 1, which was designed on page 30. Enter the first pair of data values (ie 141970, 1-59507-33132-7) and check that the expected member's name, book title and author appear automatically on the form. You will need to look at the test plan.

You can either use the drop-down lists or enter the numbers directly. The member's name should appear as soon as you exit the membership number entry box. The author and title should appear as soon as you exit the book number box. You can use the controls at the bottom left-hand corner of the form to navigate through the records that have been entered. Use the button shown below to go to a blank form ready to enter a new record.

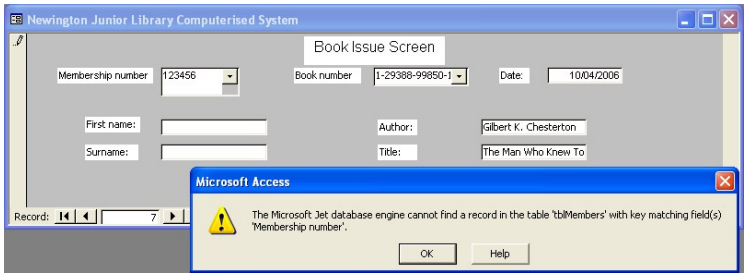
This button takes you directly to a blank form, where you can enter a new record.

4. Continue entering the test data for Test 1. Check that each pair of numbers (membership number and book number) gives the expected member's name, book title and author. When you have entered all six data pairs, close the form and open the table tblLoans in data view. You may find that the values in Loan number are different to those shown on the right. This is because you have inserted new records and then deleted them. AutoNumber produces a unique value and never reuses old values. It does not matter if your Loan number key field values do not match those shown in the diagram.



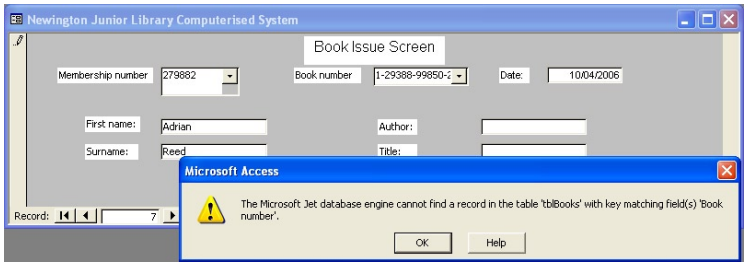
5. Now copy the screen with tblLoans in data view and obtain a 2.5 inch wide printout of the table, just like you did in steps 1 and 2 of this testing for the empty table. Stick it into Box B of the test report (page 44).

Test 2: Following test plan 2 as given on page 31

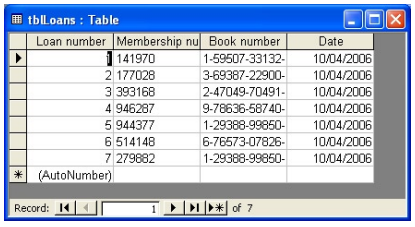
1. 

Open frmBookIssue. Click the button to enter a new record and enter the first pair of data values (ie as Test 2.1 on page 31). You should see the book name appear, but the member's name will remain blank. When you click to go onto another new record, you will see an error message appear.

Move the error message so that it is in the position shown. Do not close the form but copy the screen and paste it as before. Crop and resize your image so that it is 3 inches wide and shows the form and error message. Print the image, cut it out and stick it into Box C of the test report (page 44).

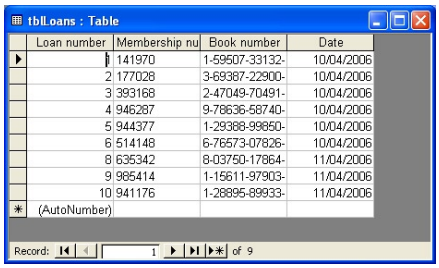
2. 

Return to Access, click OK on the error message and enter the data for Test 2.2. Take a screen shot and print the result as before. Stick your printout into Box D of the test report (page 45).



3. Now enter the data for Test 2.3. This time, when you click to go onto a new record, there is no error message. The data has been accepted and the test has failed! Obtain a screen shot of the table with the invalid data added. The image should be 2.5 inches wide. Stick your printout into Box E of the test report (page 45). When you have done this, return to Access, select the row with the invalid record and delete it.

Test 3: Following the test plan 3 given on page 31

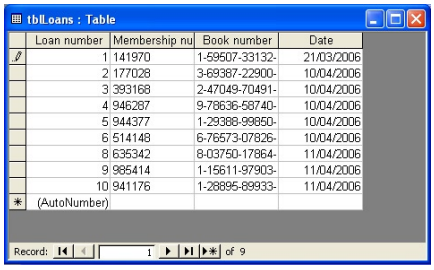


1. Check that you have deleted the invalid record added in Test 2.3. Then carry out Test 3 as given on page 31. When the test is completed, print out a screen shot of the table with the additional three records added. Your screen shot should be 2.5 inches wide. Stick your screen shot into Box F of the test report (page 45). Write a description of what the screen shot is showing. Say whether or not the test has been successful.

Implementing the address labels

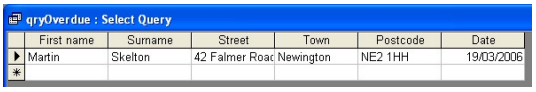
Before continuing with the testing, you need to complete the final piece of implementation. So far you have not implemented the address label design. The labels will be a mailmerge document linked to qryOverdue, ie the query that identifies the members who have overdue books.

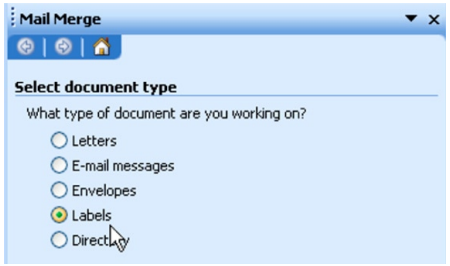
Microsoft® Word will report an error if you try to link a mailmerge document to a query that has no results so, before you create the mailmerge document, we need to set up at least one member with an overdue book.



1. Open tblLoans in data view. Change the date in the first record (for Martin Skelton borrowing *The Dawn of a Tomorrow*) to be 22 days ago. This will guarantee that there is at least one result in qryOverdue. Note that the date should be 22 days ago from today's date, not from the date the book was originally issued!

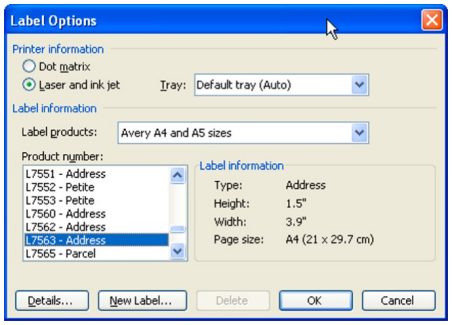
2. Check that there really is a record by opening your query qryOverdue in data view. At this point, it does not matter which records you see as long as there is at least one.

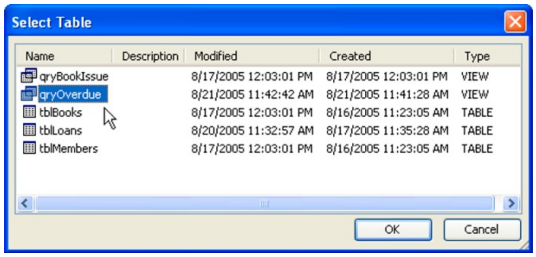




3. Close your database and open Microsoft® Word. Select Tools/Letters and Mailings/Mail Merge to open the Mail Merge wizard panel. Select Labels from the list of document types and then click Next to move on to the next stage of the wizard.

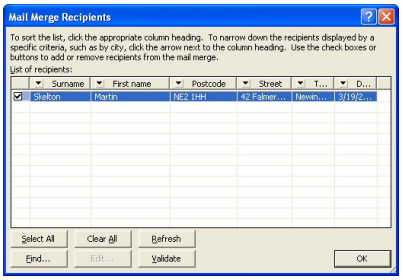
4. In step 2 of the wizard, click the Label options link to open the Label Options dialogue box. Make sure that you have the Avery A4 and A5 sizes selected in the Label products box and choose a label type that is reasonably close to your chosen size. Look back at your design diagram as required on page 29.

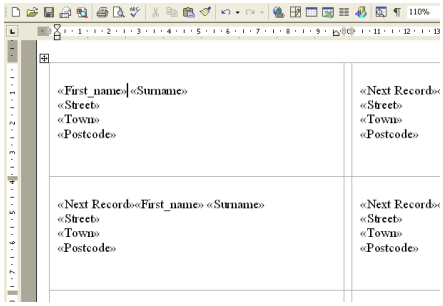




5 When you have selected a label, click OK to move on to step 3 of the wizard. Click on Browse and find and open your database file. Then select the query qryOverdue as the data source for your merge document and click OK.

6 You will see the data from your query displayed. Click OK to close this dialogue box without making any changes. Click Next to move on to step 4 of the wizard.

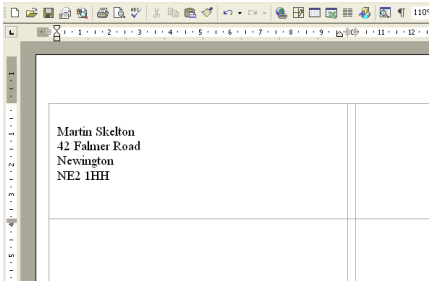




7 Use the More Items link to insert the merge fields for the top left-hand corner label according to your design. Once this label is set up correctly, click on Update all labels to copy the layout to the other labels on the page.

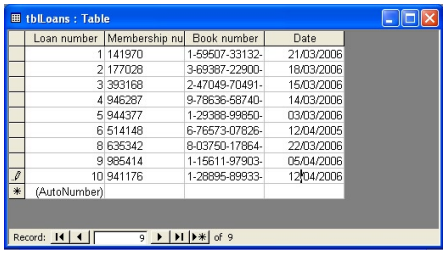
You should now have a document with 14 labels on a single page. Each label will consist of the address merge fields.

8 Click Next to move on to step 5 of the wizard, where you will see a preview of your labels. You should now see one label corresponding to the record shown by qryOverdue in step 2 of this section. Save your mailmerge document as Labels.



Testing the labels

Test 4: Following the test plan 4 as given on page 32



1 Start by filling in the dates in the last column of the table on page 32. Remember that the dates are worked out from today's date. Also, you will need to do your testing on the day that you work out the dates so if it is near the end of a lesson, calculate dates from the date of your next lesson. The spreadsheet DateCalculator.xls will work out the dates for you if you prefer (your teacher will tell you where you can load a copy of this file from). Then change the dates in tblLoans to the dates you have worked out. Print out a screen shot of tblLoans showing these dates entered. The screen shot should be 2.5 inches wide. Stick it into Box G of the test report (page 46).

2 Close the database and open your mailmerge labels document (Labels.doc). Start the mailmerge wizard and move to step 6 to perform the merge and print out the labels. There should be three if the test has been successful. Trim your (full size) printout so that it fits in Box H of the test report (page 46).

* File corrupted! Call support on 01223 350555 *

Book issue test report

Test 1

Box A

This screen shot shows the table tblLoans before any testing had been done. There are no records in the table.

Box B

This screen shot shows the six records that have been added after Test 1 has been completed.

The loan number values have been created automatically by Microsoft® Access. The date has been correctly inserted by Access and is the date on which the test was performed.

The six records contain the membership numbers and the book numbers given in the test plan for Test 1. When the test was performed, the expected members' names, book titles and authors were displayed on the input form.

The test was completed successfully and shows that the form frmBookIssue can be used to enter new records correctly into tblLoans.

Test 2

Box C

Test 2.1: The data given in test 2.1 on page 31 was entered using frmBookIssue. When I tried to go onto a new record, an error message was produced, indicating that the membership number did not exist. Microsoft® Access would not save the record. The test was successful, with the results matching the expected results described in the test plan, and showing that invalid membership numbers will not be accepted. The screen shot shows the form with the invalid membership number entered and the error message displayed.

Box D

Test 2.2: The data for Test 2.2 was entered with a valid membership number but invalid book number. The result was as expected, with the book details blank, the member's name displayed and an error message when I tried to move on to enter a new record. This is shown in the screen shot. The test was successful, showing that invalid book numbers will not be accepted.

Box E

Test 2.3: The data for Test 2.3 was entered. There was no error message when the data was saved. This was not expected and so the test has failed. This test shows that it is possible to issue the same book twice, which is invalid. The screen shot shows the table with the invalid entry on the final row.

Test 3

Box F

Test 3: _____

_____

_____

_____

_____

_____

Test 4

Box G

Test 4: The screen shot on the left shows the test data dates for Test 4 entered into tblLoans. The printout below shows the result when the labels mailmerge document is merged with query qryOverdue to produce the labels. The three labels printed are the ones expected from the test plan. The test has been successful.

Box H

Book issue evaluation

After completing all the practicals and now that the book issue system has been implemented and tested, you can complete the evaluation. Make sure that you refer to your testing, using it as evidence. Mention anything that did not work and any problems this might cause when the system is used.

Suggestions for future improvement:

* File corrupted! Call support on 01223 350555 *

Student record sheet

Week	Task	Pages	Target date	Completion date	Grade	Target/Comment
20	Members, books and CSV files	7 and 8				
21	Queries: Try it yourself	17 and 18				
E	Adding interaction to queries	18 and 19				
E	Find a book design and test plan	20 and 21				
E	Find a book implementation	21 and 22				
E	Finding a book testing	22				
E	Finding a book evaluation	22				
21	Mailmerge design and test plan	23 to 25				
23	Mailmerge implementation	26				
23	Mailmerge evaluation	26				
24	Book issue analysis	27				
24	Book issue design	28 and 29				
26	Book issue test report	30 to 32				
26	Book issue evaluation	47				

Key E = Extension work